IN THE SPECIFICATION:

Please replace paragraph [0007] with the following amended paragraph:

[0007] Some conventional debuggers support unconditional breakpoints where the execution of the program is always halted upon reaching the breakpoint. ~~While other~~ Other debuggers support conditional breakpoints that halt the execution of a program only when a predetermined value is obtained when the breakpoint is encountered.

Please replace paragraph [0029] with the following amended paragraph:

[0029] Figure 8A is a flow diagram illustrating the operation of a breakpoint manager.

Please insert the following paragraph after paragraph [0029]:

[0029.1] Figure 8B is a continuation of the flow diagram beginning in Figure 8A.

Please replace paragraph [0061] with the following amended paragraph:

[0061] One embodiment illustrating the operation of the breakpoint manager 128 is shown as a breakpoint manager routine 800 in Figure 8A and continued in Figure 8B. The routine 800 is illustrated as an event-driven routine for the breakpoint manager 128. In an event-driven system, the breakpoint manager routine 800 waits for various events. In response to an event, the breakpoint manager routine 800 executes predetermined program code to handle the event. Once the event has been handled, the breakpoint manager routine 800 returns to a state of waiting for additional events.

Please replace paragraph [0063] with the following amended paragraph:

[0063] The method 800 is entered at step 802~~0~~ where the breakpoint manager 128 receives an event. At step 804, the method queries if the event is an establish breakpoint message. If so, the method proceeds to step 806 where a new breakpoint is set. At step 808, the breakpoint count 508 in the call graph node 500 for the appropriate routine is incremented. At step 810, the operational count 600 is incremented. When the operational count 600 is incremented, this indicates that the breakpoint configuration has changed and that any on-going break point analysis

previously done will need to be re-computed. Thus, at step 812, any on-going breakpoint analysis task is stopped since the existence of a new breakpoint set in step 806 can affect the results of the analysis. At step 828, the breakpoint analysis task is restarted.